



仮想キューによる高性能ハードウェア RTOS の実現

Exploiting Virtual Queue for Implementing a High Performance RTOS in Hardware

丸山修孝
カーネロンシリコン(株)

石原 亨
九州大学

安浦 寛人
九州大学

Rev.1.0

ネットワークの高速化の進展と問題点

ネットワークの急速な普及

→ 家電、モバイル機器等民生品に普及



組込系でのTCP/IP性能の低さ

我々の実験

50MHzのARMでのTCP/IP性能
..... わずか11Mbps

400MHz (ARMの限界) でも
88Mbps

一般的な克服方法

1. CPU性能を向上させる(クロックを上げる)
 - 消費電力のが致命的
 - クロックレートの上限がTCP/IP性能の上限になる
2. TCP/IPのハードウェア化(TOE)
 - > 改修ができない(改修はLSIの作り替えを意味する)
 - > あとから機能拡張ができない
 - > TCP/IP以外がボトルネックになる

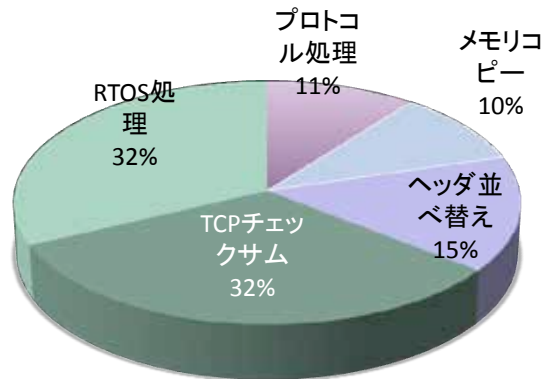
TCP/IPファームウェアの分析

市販Firmware TCP/IP実行時の各プロセスのCPU占有時間

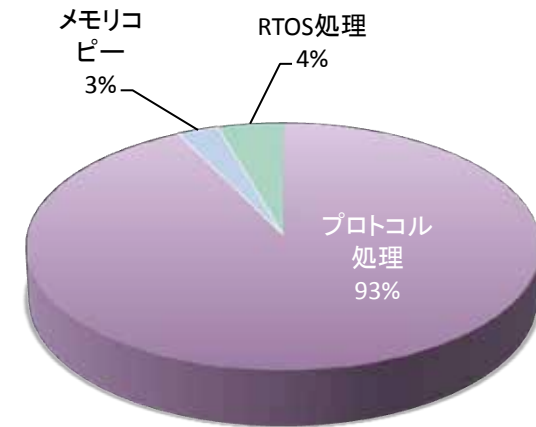


プロトコル処理は
全体の処理時間の10%程度

プロトコル処理以外の処理を
ハードウェア化し高速化



従来技術

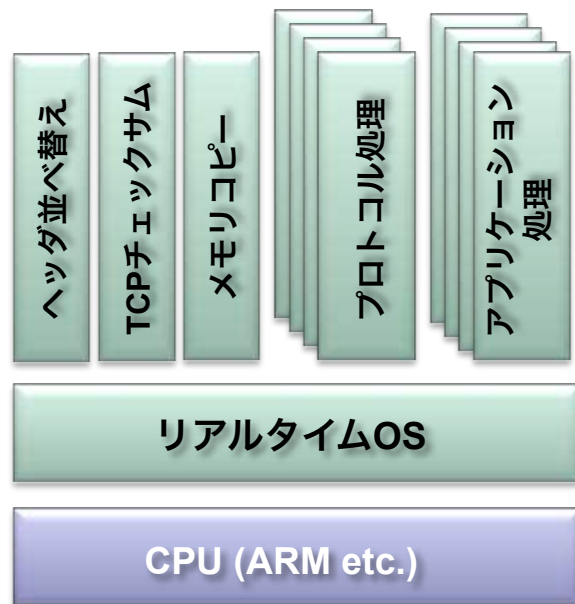


ARTESSO

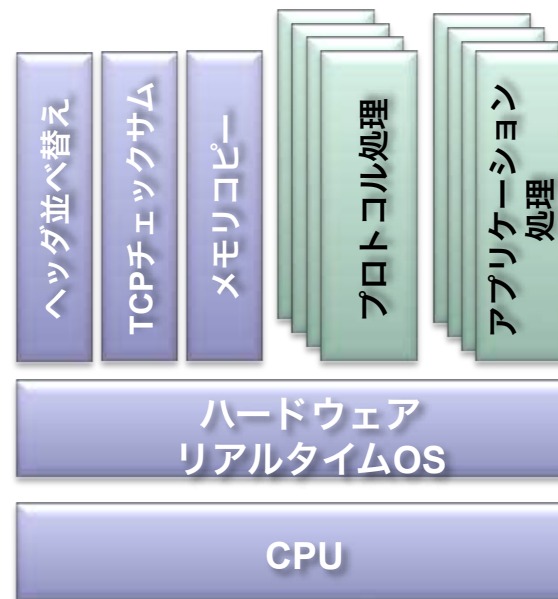
CPUがプロトコル処理に専念できる

この結果TCP/IP性能が約10倍に (消費電力は1/10)

従来の方法とARTESSOの比較



従来の方法

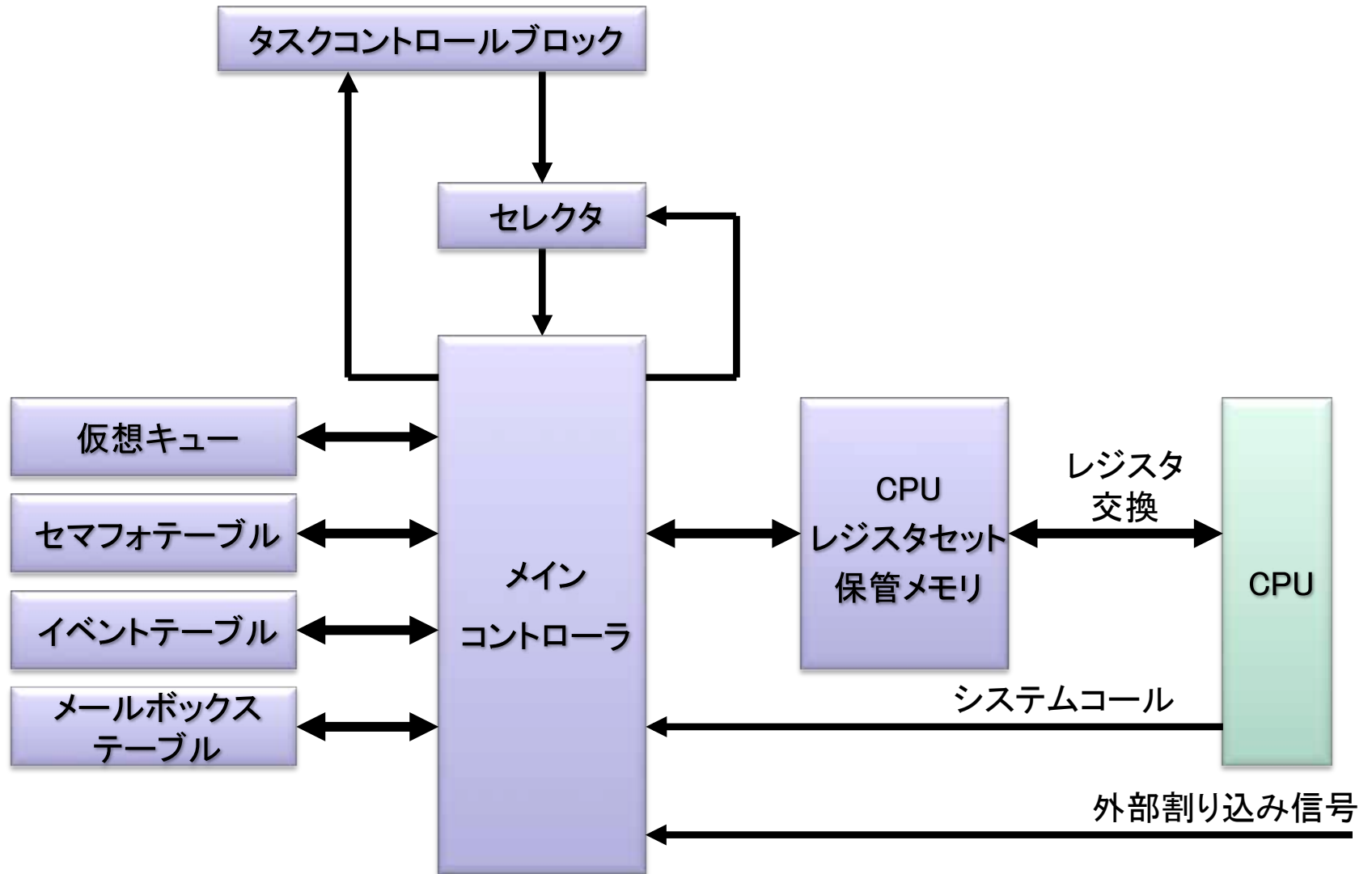


ARTESSO



Full HW TOE

ARTESSO RTOSの内部構造



キューのハードウェア化の課題

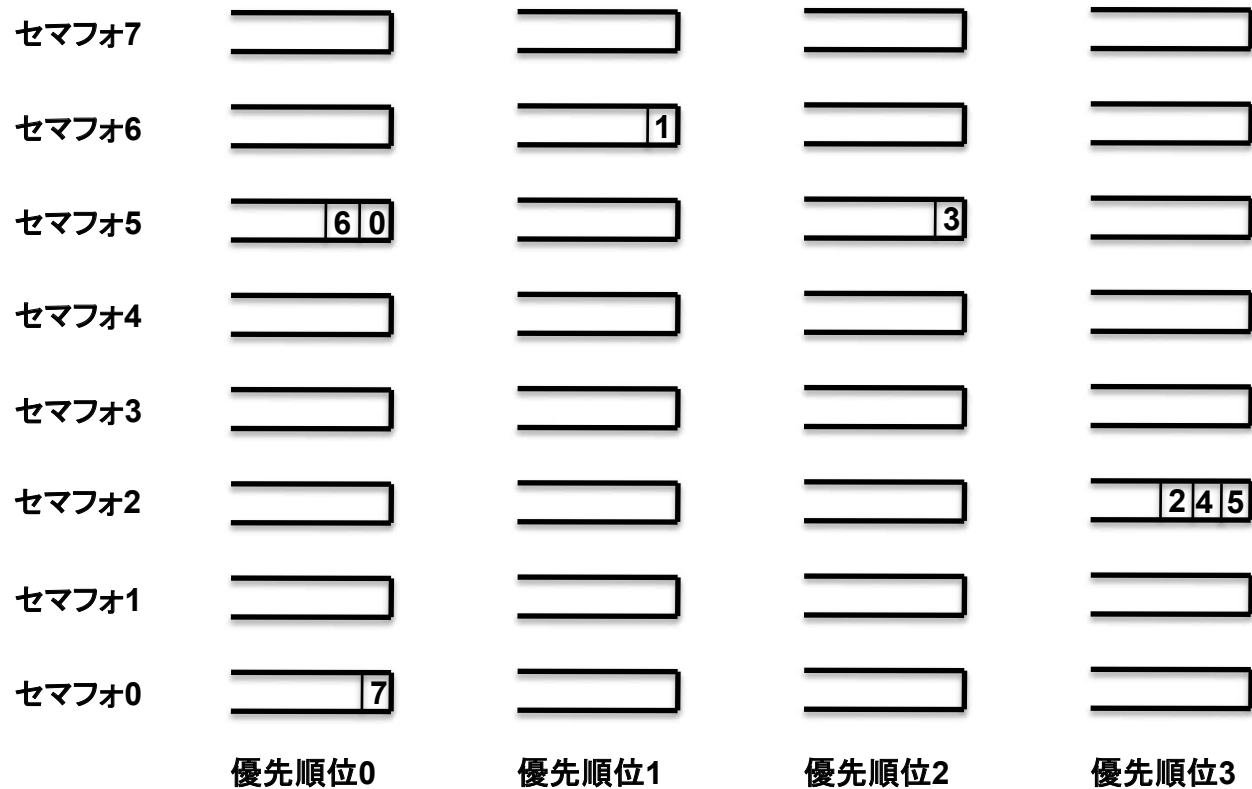
- 一般的なRTOS……数千個のキューを内在
 - 例) 32セマフォ、32イベント、192メールボックス、16優先順位
 - $(32 + 32 + 192) \times 16 = 4,096$ 個のキューが必要
 - ⇒ FIFOで実現すると巨大な回路になってしまう
- RTOSではキューの途中から取り出す機能が必要
 - 例) セマフォ待ちでタイムアウトした場合や、
 - 「待ち状態の強制解除」システムコールが発行された場合
 - ⇒ この機能はFIFOでは実現できない

従来のキュー

セマフォ待ちのキューの例

セマフォの数8、優先順位4、タスクの数8

待ち行列の中に書き込まれているのがタスク識別子



問題点…… (セマフォの数×優先順位の数) キューが必要
 全てのキューはタスクの数だけの容量が必要。

仮想キューとは？

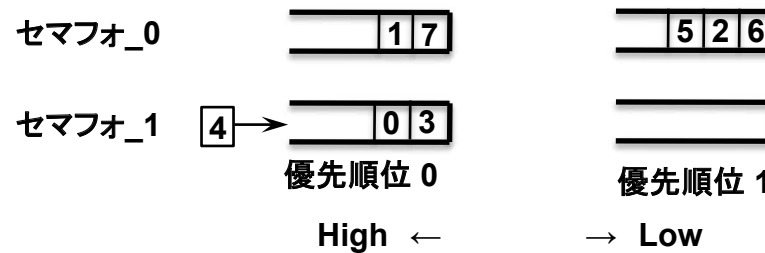
所属しているキュー識別子	セマフォ5	セマフォ6	セマフォ2	セマフォ5	セマフォ2	セマフォ2	セマフォ5	セマフォ0
優先順位	0	1	3	2	3	3	0	0
キューに入った順序	5	3	0	7	4	6	2	1
	タスク0 レジスタ	タスク1 レジスタ	タスク2 レジスタ	タスク3 レジスタ	タスク4 レジスタ	タスク5 レジスタ	タスク6 レジスタ	タスク7 レジスタ

● 仮想キュー …… キュー情報の圧縮技術

- 従来のキュー …… キュー毎にタスクの情報を保持
- 仮想キュー …… タスク毎にキュー情報を管理

仮想キューの動作 (1)

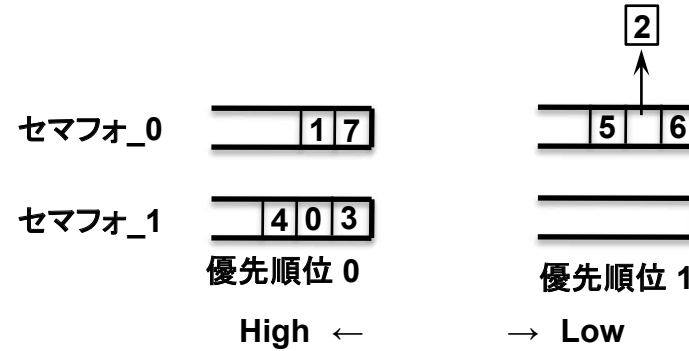
エンキュー動作



	Task 0 Register	Task 1 Register	Task 2 Register	Task 3 Register	Task 4 Register	Task 5 Register	Task 6 Register	Task 7 Register
タスク識別子	0	1	2	3	4	5	6	7
キュー識別子	S_1	S_0	S_0	S_1	S_1	S_0	S_0	S_0
優先順位	0	0	1	0	0	1	1	0
順序	4 ⇒ 5	1 ⇒ 2	3 ⇒ 4	6 ⇒ 7	0	0 ⇒ 1	5 ⇒ 6	2 ⇒ 3

仮想キューの動作 (2)

検索削除動作



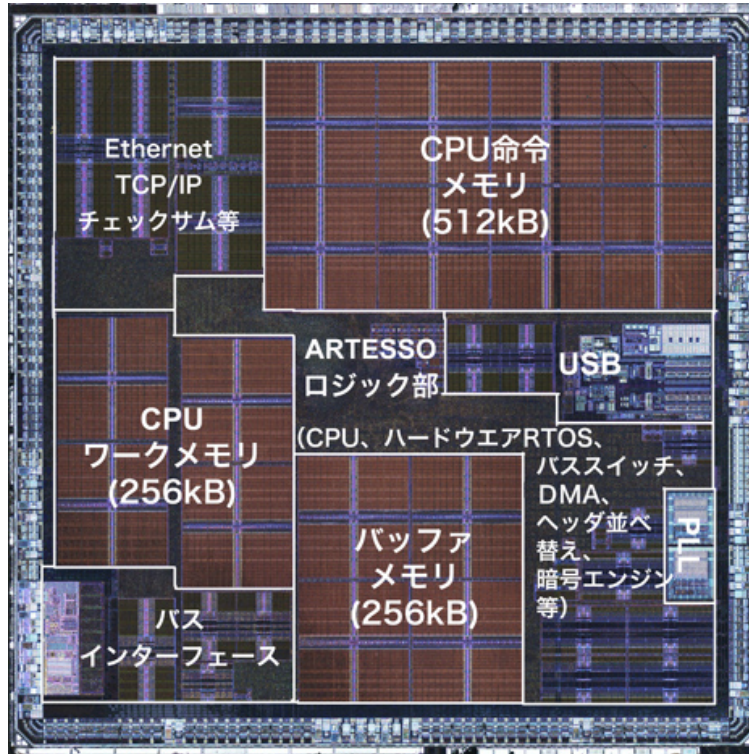
	Task 0 Register	Task 1 Register	Task 2 Register	Task 3 Register	Task 4 Register	Task 5 Register	Task 6 Register	Task 7 Register
タスク識別子	0	1	2	3	4	5	6	7
キュー識別子	S_1	S_0	Non	S_1	S_1	S_0	S_0	S_0
優先順位	0	0	Non	0	0	1	1	0
順序	5 ⇒ 4	2	4 ⇒ Non	7 ⇒ 6	0	1	6 ⇒ 5	3

ARTESSO RTOSの性能

各システムコールに必要なクロック数

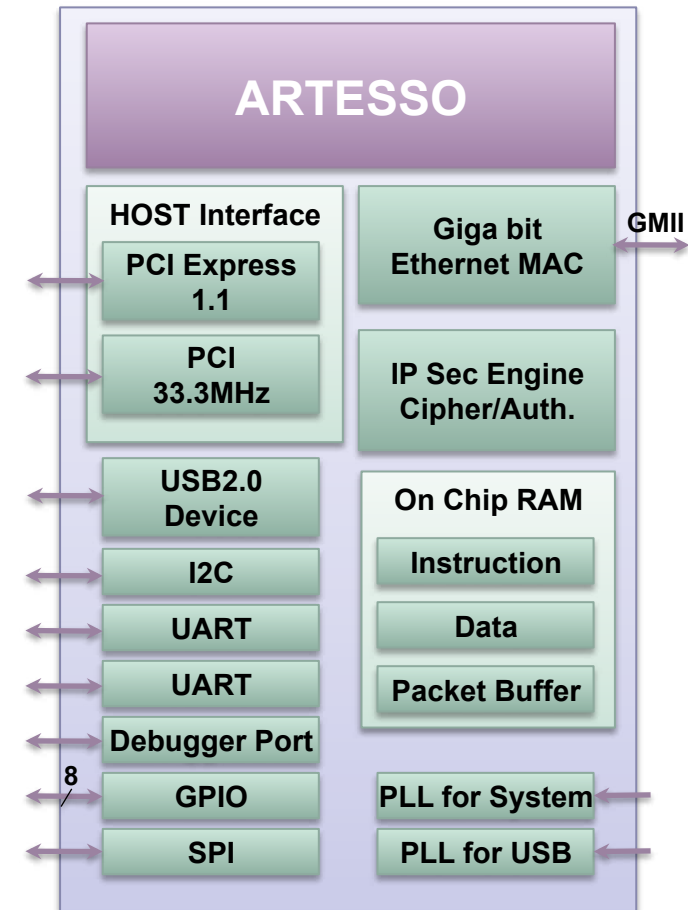
システムコール名	ディスパッチ	NORTi (ARM926)	ARTESSO V4.2
起床待ち/tslp_tsk	有	628	10
タスクの起床/wup_tsk	有	496	10
メールボックスからの受信/rcv_mbx	無	224	7
メールボックスからの受信/rcv_mbx	有	591	11
メールボックスへの送信/snd_mbx	無	360	8
メールボックスへの送信/snd_mbx	有	541	11
セマフォ待ち/wai_sem	無	216	6
セマフォ待ち /wai_sem	有	558	9
セマフォ 返却/sig_sem	無	344	7
セマフォ 返却/sig_sem	有	536	11

LSIによる試作



90nmプロセスASIC

動作クロック150MHzで
TCP/IPスループット515Mbpsを達成



ブロック図

ASICの各要素の面積

	Area (μm ²)
全面積	27,601,284
命令、データ、バッファ用RAM	14,851,332
ARTESSO	2,919,025
RTOS	540,170
ロジック	372,964
仮想キュー	30,213
その他	342,751
RAM	167,206
その他 (CPU, ヘッダ並べ替え, DMA, 暗号化, etc.)	2,378,855
その他 (Ethernet, USB, Bus Interface)	9,830,927

90nmプロセスASIC

● RTOSの仕様

- 32 タスク
- 32 セマフォ識別子数
- 32 イベント識別子数
- 192 メールボックス識別子数
- 16 優先順位

● ウェイトキューの総計 4,096.

● ゲート数

- RTOS 170,000
- Virtual Queue 38,300

結 論

- TCP/IP等のプロトコル処理はCPU処理の多くの時間をRTOS処理に消費する
- したがってRTOSをハードウェア化し、高速化することによりTCP/IP処理を高速化することができる。
- RTOSのハードウェア化において大きな障害は、RTOSが多量にキューを使用するということである。
- 「仮想キュー」技術により少ないハードウェア量で多量のキューを実装することが可能
- ネットワーク処理以外、例えば高度なリアルタイム性を必要とする自動車のマシン制御等への応用も可能

